AFRL-VA-WP-TP-2002-331

STABLE NEURAL CONTROL OF UNCERTAIN MULTIVARIABLE SYSTEMS

Mark J. Mears
Marios M. Polycarpou

DECEMBER 2001

Approved for public release; distribution is unlimited.

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
December 2001	Journal Article Preprint	

4. TITLE AND SUBTITLE
STABLE NEURAL CONTROL OF UNCERTAIN MULTIVARIABLE SYSTEMS

5a. CONTRACT NUMBER
IN-HOUSE

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)
Mark J. Mears (AFRL/VACA)
Marios M. Polycarpou (University of Cincinnati)

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Control Theory Optimization Branch (AFRL/VACA)
Control Sciences Division
Air Vehicles Directorate
Air Force Research Laboratory, Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

University of Cincinnati
Dept. ECECS
Cincinnati, OH
45221-0030

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-VA-WP-TP-2002-331

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2002-331

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Journal preprint for *International Journal of Adaptive Control and Signal Processing,* January 18, 2002.

14. ABSTRACT ***(Maximum 200 Words)***
Tracking control of a class of nonlinear, uncertain, multi-input, multiple-output systems is addressed in this paper. The control system architecture uses neural networks for function approximation, certainty equivalent control inputs to cancel plant dynamics and smoothed sliding mode control to insure that the trajectories remain bonded. Lyapunov analysis is used to derive equations for the sliding mode control, neural network training, and to show uniform ultimate boundedness of the closed loop systems. Stability analysis results are shown for single-input single-output and two-input two-output systems. Results are then extended to the more general multiple-input multiple-output case where the number of inputs is equal to the number of outputs. Simple simulation examples are used to illustrate control system performance.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mark Mears
Unclassified	Unclassified	Unclassified	SAR	30	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-8685

Stable Neural Control of Uncertain Multivariable Systems

Mark J. Mears[1,2] and Marios M. Polycarpou[2]

[1]Wright-Patterson Air Force Base, AFRL/VACA, WPAFB, OH 45433-7521, USA

[2]Dept. ECECS, University of Cincinnati, Cincinnati, OH 45221-0030, USA

December 21, 2001

Abstract

Tracking control of a class of nonlinear, uncertain, multi-input, multiple-output systems is addressed in this paper. The control system architecture uses neural networks for function approximation, certainty equivalent control inputs to cancel plant dynamics and smoothed sliding mode control to insure that the trajectories remain bounded. Lyapunov analysis is used to derive equations for the sliding mode control, neural network training, and to show uniform ultimate boundedness of the closed loop system. Stability analysis results are shown for single-input single-output and two-input two-output systems. Results are then extended to the more general multiple-input multiple-output case where the number of inputs is equal to the number of outputs. Simple simulation examples are used to illustrate control system performance.

1 Introduction

The goal of this work is to apply intelligent control methods to high performance manned aircraft and Unmanned Air Vehicles (UAVs) [9, 21]. These systems are characterized by multiple input/output variables, large nonlinearities, significant uncertainties, configuration variations, and planform changes through decades of operation. Through all this, the control system is required to maintain the highest levels of performance, guarantee stability and robustness, and to meet very stringent reliability requirements. To meet these stability and performance requirements, control

designers typically require very accurate models. Therefore, a large amount of time and effort is needed to reduce the plant's dynamic model uncertainties to tolerable levels. Some research work has focused on reducing modeling uncertainty and associated costs by developing control system design methods that use available model information generated off-line to define the nominal control system, and then use gathered data and known relationships between parameters to refine the control system on-line. In this way, some stability and control performance requirements can be met while reducing the overall system design costs [3].

Flight control system designs have historically relied on linear time invariant (LTI) models of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

where x, y, and u are the state vector, output vector and control vector respectively. The A, or *stability* matrix is based upon stability derivatives and B, or *control* matrix is based upon the control derivatives. The C is the output matrix based on how states are combined to form outputs.

The A and B matrices also depend on nominal air speed and air density. For this reason they can vary greatly from one flight condition to another, so designs are typically carried out at a number of points in the flight envelope and then blended together [18]. The performance of these control systems, which are referred to as *Gain Scheduled Controllers*, is dependent on the accuracy of a small perturbation model. Historically, gain scheduled control has worked relatively well because aircraft dynamics are predominantly a function of states such as Mach, altitude, and angle of attack. These matrices are very seldom represented as a function of the controls themselves because the controls respond faster than states and control effectiveness (represented by B) is predominantly independent of the control input, u.

In this work, we directly address the plant nonlinearities by representing the plant using a form which includes nonlinear functions of the state and which is affine with respect to the control. We also assume that we have full-state measurement, removing the need for nonlinear observers, which makes the problem considerably less complicated. The resulting generic model form is well known and given by $\dot{x} = f(x)+g(x)u$. A variety of approaches to control system design have been proposed to directly address system nonlinearities. These include feedback linearization [12], dynamic inversion [27], sliding mode control [14], and backstepping [16]. Feedback linearization transforms the plant into a linear (or partially linear) system by a suitable nonlinear state transformation and cancellation of the nonlinearities by feedback control. Backstepping is similar to feedback linearization, but does not rely on model inversion and does not require the designer to cancel helpful nonlinearities. Further,

backstepping is able to deal with plants where certain matching conditions are not satisfied.

Both feedback linearization and backstepping rely on cancellation of known nonlinearities. To address the issue of uncertainty, several techniques have been developed: (i) *adaptive* methods deal with parametric uncertainty [13, 26, 33], where the nonlinearities are assumed to be known but some of the parameters that multiply these nonlinearities are unknown or uncertain; (ii) *robust* methods deal with the case where known upper bounds on the unknown nonlinearities are available [1, 5] and therefore, they tend to be conservative, sometimes leading to high-gain feedback; (iii) robust adaptive methods combine parametric uncertainty and unknown nonlinearities with partially known bounds [24, 35].

The above control techniques are based on the assumption that the plant nonlinearities are either known or can be bounded by some known functions. In many applications, including control of high performance aircraft systems, some of the nonlinearities need to be approximated on-line. This may be due to modeling errors during the identification/modeling phase or, quite often, due to time-variations in the dynamics as a result of changes in the operating conditions or due to component wear or damage. To address the issue of unknown nonlinearities, various control system architectures have incorporated neural networks as on-line approximators of unknown nonlinearities [34]. These control systems are often labeled *connectionist*, *intelligent* or *neural*. Early work in neural control attempted to define the role of the neural networks within the control system [20] and much of this work dealt largely with the role of the network, the distinction between *adaptive* and *neural* control, and learning paradigms [8]. While research stressing these topics justifiably continues, some have been driven by application requirements to stress closed-loop system stability. Such is the case with flight control system design.

This stability focused work on neural control uses networks as a component in a mathematical framework from which adaptive control laws can be derived and stability guarantees can be made. Because networks are typically nonlinear, they are often used in combination with nonlinear control methods and Lyapunov analysis is commonly the stability analysis tool employed. Typically, the feedback control law and the adaptive law for updating the network weights are derived by utilizing a Lyapunov function, whose time derivative is forced to have some desirable stability properties (for example, negative definiteness). Therefore, the stability of the closed-loop system is obtained during the synthesis of the adaptive control laws. Examples of this type of approach, which is referred to as Lyapunov synthesis method, include [2, 4, 6, 7, 15, 19, 22, 23, 25, 28, 29, 32, 30, 36].

Most of the results in neural control based on the Lyapunov synthesis method are derived for systems with a single control input. The multivariable problem, especially if the control inputs are multiplied by unknown nonlinearities, becomes more challenging due to the coupling between

control inputs. In this paper we present a control design approach which can be applied to a class of nonlinear, affine, Multiple-Input Multiple-Output (MIMO) plants with full-state measurement, where the number of inputs is the same as the number of outputs. The control objective is to achieve tracking of some desired state trajectories. An adaptive bounding technique is employed to handle the unknown network reconstruction error approximation [19, 25]. Sufficient stabilizability conditions on the unknown control multiplier functions are derived and both neural control components and sliding mode control signals are combined in proportions determined by the accuracy of the function approximation. The Lyapunov synthesis approach is used to derive a neural control system with guaranteed stability properties.

This paper is organized as follows: We first formulate the Single-Input Single-Output (SISO) problem in Section 2 by showing the plant model structures to be considered, define some of the parameters and assumptions, and define the control structure. Analytical results are then used to show stability properties of the proposed scheme. A SISO example is provided in Section 3 to illustrate the concepts. Section 4 extends the definitions, assumptions, control architecture and stability analysis to a class of Two-Input Two-Output (TITO) systems and an illustrative example for this is shown in Section 5. The control system design is then extended to MIMO systems of arbitrary size in Section 6 and conclusions are stated in Section 7.

2 Neural Control of a SISO System

We start by considering the neural control problem for a SISO system. This illustrates some of the issues that arise in dealing with unknown nonlinearities multiplying the control input and provides a convenient framework for the extension into the MIMO domain.

Consider the problem of tracking control for a SISO plant given by

$$\dot{x} = w_1(x) + w_2(x)u \tag{1}$$

where the measurable state variable $x(t)$ belongs to a domain of interest $\Omega \subset \mathbb{R}$, $u \in \mathbb{R}$ is the control input and $w_1(x), w_2(x)$ are unknown nonlinearities which are assumed to be locally Lipschitz (to assure uniqueness of solutions). The control objective is to design a feedback control law such that $x(t)$ follows some desired trajectory $x_d(t)$ as closely as possible.

It is often the case that a control designer has a rough estimate of the characteristics of the plant dynamics either through analytical modeling or through empirical studies. The control approach being used here directly allows for this type of information to be used by having $w_1(x), w_2(x)$ each be composed of a known part and by an unknown part [23, 25]. However, for notational simplification, in this paper the functions in equation (1) are assumed to be completely unknown.

Since $w_1(x)$ and $w_2(x)$ are unknown, they are approximated for use in the control system. In this work we use linearly parameterized approximators of the form

$$\hat{w}_i(x) = \theta_{w_i}^T \xi(x) \quad \text{for } i = 1, 2 \tag{2}$$

where $\hat{w}_1(x)$ and $\hat{w}_2(x)$ are estimates of the unknown functions, $\theta_{w_1}, \theta_{w_2} \in \mathbb{R}^m$ are the parameter vector estimates and $\xi(x) : \mathbb{R} \mapsto \mathbb{R}^m$ is a vector of strictly positive basis functions. For example, Radial Basis Functions (RFB) networks using Gaussian functions can be used in this framework. As we will see later on, the assumption that each element of $\xi(x)$ is positive is used to deal with a stabilizability problem. In general, a different set of basis functions $\xi_{w_1}(x), \xi_{w_2}(x)$ could be used in approximating $w_1(x)$ and $w_2(x)$ respectively. However, in order to simplify the notation (and without any loss of generality) we let $\xi(x) = \xi_{w_1}(x) = \xi_{w_2}(x)$.

For each $i = 1, 2$ the *best* approximation for $w_i(x)$ is defined as

$$w_i^*(x) = \theta_{w_i}^{*T} \xi(x) \quad \text{for } i = 1, 2 \tag{3}$$

where $\theta_{w_i}^*$ denotes the parameter vector that minimizes the difference between $w_i(x)$ and $\theta_{w_i}^T \xi(x)$ for all x in the domain of interest, Ω, i.e.,

$$\theta_{w_1}^* = \arg \min_{\theta_{w_1} \in \mathbb{R}^m} \{\sup_{x \in \Omega} |w_1(x) - \theta_{w_1}^T \xi(x)|\} \tag{4}$$

and

$$\theta_{w_2}^* = \arg \min_{\theta_{w_2} \in \Theta_{w_2}} \{\sup_{x \in \Omega} |w_2(x) - \theta_{w_2}^T \xi(x)|\} \tag{5}$$

where $\Theta_{w_2} \subset \mathbb{R}^m$ is the set of all θ_{w_2} which lie within parameter bounds defined later. We use these *best* possible parameters given by equations (4) and (5) to define the parameter vector estimation errors as

$$\tilde{\theta}_{w_i} = \theta_{w_i} - \theta_{w_i}^* \qquad \text{for } i = 1, 2. \tag{6}$$

It is important to note that unless the actual functions, $w_i(x)$, are linear combinations of the $\xi(x)$ basis functions, there will be some residual errors remaining in each approximation even if the *best* θ_{w_i} vector is used. In this work, bounds on the approximation errors of $\hat{w}_1(x)$ are addressed using an adaptive bounding method while $\hat{w}_2(x)$ has fixed bounds to insure well-defined controls.

The error which remains after the best fit has been achieved is referred to as the *reconstruction error* and is given by

$$\delta_{w_i}(x) = w_i(x) - w_i^*(x). \tag{7}$$

The reconstruction error $\delta_{w_i}(x)$ is an important quantity in neural control design methods since it provides a measure of how close an approximation can be achieved by a given neural network architecture. In general, as the number of nodes in the network increase, the reconstruction error becomes smaller. In the limit, as the number of nodes becomes infinitely large (and centers are appropriately placed), the *universal approximation theorem* states that the reconstruction error converges to zero (over a compact domain) [10]. Unfortunately, universal approximation results are relevant only if the number of nodes start becoming very large. Therefore, nonzero reconstruction error is something that needs to be dealt with in practical applications. In this work, we define an upper bound (which is assumed to be unknown) on the magnitude of $\delta_{w_i}(x)$ as given by

$$\psi^*_{w_i} = \sup_{x \in \Omega} |\delta_{w_i}(x)|. \tag{8}$$

In the approximation of $w_1(x)$, the necessity to assume *a priori* knowledge of a bound on the reconstruction error is removed by developing an adaptive bounding scheme where the bound on $\delta_{w_1}(x)$ is estimated on-line. We define this estimate of the reconstruction error bound as $\psi_{w_1}(t)$, and the bounding estimation error, $\tilde{\psi}_{w_1}(t)$, which will be used in the stability analysis is defined as

$$\tilde{\psi}_{w_1}(t) = \psi_{w_1}(t) - \bar{\psi}^*_{w_1}. \tag{9}$$

where $\bar{\psi}^*_{w_1} = \max\{\psi^*_{w_1}, \psi^o_{w_1}\}$ and $\psi^o_{w_1}$ is a design parameter that will appear in the adaptation law for updating $\psi_{w_1}(t)$.

Next, we define the tracking error, $e(t) = x(t) - x_d(t)$, which is used to obtain the sliding mode scalar, s, where

$$s(t) = e(t) + c\int_0^t e(\tau)\, d\tau, \tag{10}$$

and where c is a positive design constant. As we will see later on, $s(t)$ is used to generate the feedback control law, as well as the adaptive laws for updating the network weights. By defining the sliding mode quantity in this way, we obtain a filtered error which induces integral control action [17, 31].

The parameter vectors, θ_{w_1} and θ_{w_2}, and the adaptive bound, ψ_{w_1}, are updated according to the following adaptive laws:

$$\dot{\theta}_{w_1} = \gamma\{s\xi - \sigma(\theta_{w_1} - \theta^o_{w_1})\} \tag{11}$$

$$\dot{\theta}_{w_2} = \gamma u s \xi \tag{12}$$

$$\dot{\psi}_{w_1} = \gamma\{s\,\tanh(\frac{s}{\epsilon}) - \sigma(\psi_{w_1} - \psi^o_{w_1})\} \tag{13}$$

where $\theta^o_{w_1}$ and $\psi^o_{w_1}$, are design parameters (representing *a priori* estimates of the unknown network parameters and adaptive bound) and $\sigma > 0$ is a leakage constant used in the framework of the σ-modification [11] to prevent the parameter estimates from exhibiting parameter drift. The constant

$\gamma > 0$ denotes the adaptation gain, and the $\epsilon > 0$ is a small design constant used to smooth out the sign function by converting it into a hyperbolic function (tanh). As shown in the subsequent analysis, these update laws are derived based on the Lyapunov synthesis approach to maintain stability and prevent parameter drift.

The network reconstruction error associated with the $w_2(x)$ function is treated differently than the one associated with the $w_1(x)$. Typically, there is more a priori information about the range of possible values for $w_2(x)$. Furthermore, the adaptive bounding technique developed for $w_1(x)$ (by the use of the adaptive parameter $\psi_{w_1}(t)$) cannot be directly applied to the case of $w_2(x)$ due to a stabilizability problem. Therefore, it is assumed that upper and lower bounds on $w_2(x)$ are available such that $\overline{w}_2(x) > w_2(x) > \underline{w}_2(x) > 0 \ \forall x \in \Omega$. These upper and lower bounds are chosen such that there exist parameter vectors $\overline{\theta}_{w_2}$ and $\underline{\theta}_{w_2}$ satisfying $\overline{w}_2(x) = \overline{\theta}_{w_2}^T \xi(x)$ and $\underline{w}_2(x) = \underline{\theta}_{w_2}^T \xi(x)$. The parameter vectors $\overline{\theta}_{w_2}, \underline{\theta}_{w_2}$ are then used to bound the parameter estimate vector θ_{w_2}. It is worth noting that this approach can also be applied to systems where $w_2(x) < 0 \ \forall x \in \Omega$ by changing the input sign convention.

Using the previously defined quantities, we are now ready to define the feedback control signal as

$$u = \frac{u_a}{\theta_{w_2}^T \xi + \psi_{w_2}^* \operatorname{sgn}(u_a s)} \tag{14}$$

where u_a is defined as

$$u_a = \dot{x}_d - ce - ks - \theta_{w_1}^T \xi - \psi_{w_1} \tanh(\frac{s}{\epsilon}) \tag{15}$$

and k is a positive design constant. As we will see later, the adaptive law for θ_{w_2} will be modified to ensure that the denominator of the control equation is bounded away from zero.

We begin the stability analysis by defining the Lyapunov function as

$$V(s, \tilde{\theta}_{w_i}, \tilde{\psi}_{w_1}) = \frac{1}{2}s^2 + \frac{1}{2\gamma}\sum_{i=1}^{2} \tilde{\theta}_{w_i}^T \tilde{\theta}_{w_i} + \frac{1}{2\gamma}\tilde{\psi}_{w_1}^2 \tag{16}$$

where s is given by equation (10), $\tilde{\psi}_{w_1}(t)$ is given by equation (9), and $\tilde{\theta}_{w_i}$ is given by equation (6). Taking the derivative of V with respect to time, substituting for $\dot{s}$ and adding a zero sum term we get

$$\dot{V} = s\{w_1 + w_2 u - \dot{x}_d + ce + (u_a - u_a)\} + \frac{1}{\gamma}\sum_{i=1}^{2} \tilde{\theta}_{w_i}^T \dot{\theta}_{w_i} + \frac{1}{\gamma}\tilde{\psi}_{w_1}\dot{\psi}_{w_1}. \tag{17}$$

Substituting (15) and (7) into (17) we get

$$\begin{aligned}\dot{V} &= s\{(w_1 + w_2 u - ks - \theta_{w_1}^T \xi - \psi_{w_1} \tanh(\frac{s}{\epsilon}) - u_a\} + \frac{1}{\gamma}\{\sum_{i=1}^{2} \tilde{\theta}_{w_i}^T \dot{\theta}_{w_i} + \tilde{\psi}_{w_1}\dot{\psi}_{w_1}\} \\ &= -ks^2 + s\{\delta_{w_1} - \tilde{\theta}_{w_1}^T \xi + (w_2 u - u_a) - \psi_{w_1} \tanh(\frac{s}{\epsilon})\} \\ &\quad + \frac{1}{\gamma}\{\sum_{i=1}^{2} \tilde{\theta}_{w_i}^T \dot{\theta}_{w_i} + \tilde{\psi}_{w_1}\dot{\psi}_{w_1}\}. \end{aligned} \tag{18}$$

Then by using $\bar{\psi}_{w_1}^*$ to bound δ_{w_1} and using equation (9) we can obtain an upper bound on $\dot{V}$ as follows:

$$\begin{aligned}\dot{V} &\leq -ks^2 + |s|\bar{\psi}_{w_1}^* - s\bar{\psi}_{w_1}^* \tanh(\frac{s}{\epsilon}) - s\tilde{\psi}_{w_1} \tanh(\frac{s}{\epsilon}) - s\tilde{\theta}_{w_1}^T \xi + s(w_2 u - u_a)\} \\ &\quad + \frac{1}{\gamma}\{\sum_{i=1}^{2} \tilde{\theta}_{w_i}^T \dot{\theta}_{w_i} + \tilde{\psi}_{w_1}\dot{\psi}_{w_1}\}. \end{aligned} \tag{19}$$

To bound the terms involving $\bar{\psi}_{w_1}^*$, we use a property of the hyperbolic tangent [23] according to which for any $\nu \in \mathbb{R}$ and any constant $\epsilon > 0$

$$0 \leq |\nu| - \nu \tanh(\frac{\nu}{\epsilon}) \leq \kappa\epsilon \tag{20}$$

where $\kappa = 0.2786$.

By using equations (11), (13) and (20) in (19), we can write the inequality as

$$\begin{aligned}\dot{V} &\leq -ks^2 + \kappa\epsilon\bar{\psi}_{w_1}^* - \sigma\tilde{\psi}_{w_1}(\psi_{w_1} - \psi_{w_1}^o) - \sigma\tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) + s(w_2 u - u_a)\} \\ &\quad + \frac{1}{\gamma}\tilde{\theta}_{w_2}^T \dot{\theta}_{w_2}. \end{aligned} \tag{21}$$

Since the denominator of the control law described by (14) will be designed such that it remains positive for all $t \geq 0$, it can be readily concluded that $sgn(u_a s) = sgn(us)$. Now, by substituting (14) into (21), we obtain

$$\begin{aligned}\dot{V} &\leq -ks^2 + \kappa\epsilon\bar{\psi}_{w_1}^* - \sigma\tilde{\psi}_{w_1}(\psi_{w_1} - \psi_{w_1}^o) - \sigma\tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) \\ &\quad + su\{w_2 - [\theta_{w_2}^T \xi + \psi_{w_2}^* \operatorname{sgn}(us)]\} + \frac{1}{\gamma}\tilde{\theta}_{w_2}^T \dot{\theta}_{w_2} \\ &\leq -ks^2 + \kappa\epsilon\bar{\psi}_{w_1}^* - \sigma\tilde{\psi}_{w_1}(\psi_{w_1} - \psi_{w_1}^o) - \sigma\tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) + su\delta_{w_2} - |su|\psi_{w_2}^* \\ &\quad - su\tilde{\theta}_{w_2}^T \xi + \frac{1}{\gamma}\tilde{\theta}_{w_2}^T \dot{\theta}_{w_2}. \end{aligned} \tag{22}$$

Now using equation (12) and knowing that $\psi_{w_2}^* > \delta_{w_2}$ we arrive at a bound on the Lyapunov derivative given by

$$\dot{V} \leq -ks^2 + \kappa\epsilon\bar{\psi}_{w_1}^* - \sigma\{\tilde{\psi}_{w_1}(\psi_{w_1} - \psi_{w_1}^o) + \tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o)\}. \tag{23}$$

By completing the squares it can be shown that

$$\begin{aligned}\dot{V} \leq & -ks^2 + \kappa\epsilon\bar{\psi}^*_{w_1} - \frac{\sigma}{2}\{\tilde{\theta}^T_{w_1}\tilde{\theta}_{w_1} + (\theta_{w_1} - \theta^o_{w_1})^T(\theta_{w_1} - \theta^o_{w_1}) - (\theta^*_{w_1} - \theta^o_{w_1})^T(\theta^*_{w_1} - \theta^o_{w_1})\} \\ & \frac{\sigma}{2}\{\tilde{\psi}_{w_1}\tilde{\psi}_{w_1} + (\psi_{w_1} - \psi^o_{w_1})(\psi_{w_1} - \psi^o_{w_1}) - (\psi^*_{w_1} - \psi^o_{w_1})(\psi^*_{w_1} - \psi^o_{w_1})\}. \end{aligned} \quad (24)$$

Because $(\theta_{w_1} - \theta^o_{w_1})^T(\theta_{w_1} - \theta^o_{w_1}) \geq 0$ and $(\psi_{w_1} - \psi^o_{w_1})(\psi_{w_1} - \psi^o_{w_1}) \geq 0$, equation (24) implies that

$$\begin{aligned}\dot{V} \leq & -ks^2 + \kappa\epsilon\bar{\psi}^*_{w_1} - \frac{\sigma}{2}\tilde{\theta}^T_{w_1}\tilde{\theta}_{w_1} + \frac{\sigma}{2}\{(\theta^*_{w_1} - \theta^o_{w_1})^T(\theta^*_{w_1} - \theta^o_{w_1})\} \\ & -\frac{\sigma}{2}\tilde{\psi}^T_{w_1}\tilde{\psi}_{w_1} + \frac{\sigma}{2}\{(\psi^*_{w_1} - \psi^o_{w_1})^T(\psi^*_{w_1} - \psi^o_{w_1})\}. \end{aligned} \quad (25)$$

We would like to be able to make stability conclusions about the system by putting the system into the form of

$$\dot{V} \leq -bV + \lambda \quad (26)$$

where $b \in \mathbb{R}^+$ and λ is a group of constant terms. To satisfy (26) it suffices to insure that

$$\begin{aligned}-ks^2 + \kappa\epsilon\bar{\psi}^*_{w_1} - \frac{\sigma}{2}\tilde{\theta}^T_{w_1}\tilde{\theta}_{w_1} + \frac{\sigma}{2}\{(\theta^*_{w_1} - \theta^o_{w_1})^T(\theta^*_{w_1} - \theta^o_{w_1})\} - \frac{\sigma}{2}\tilde{\psi}^T_{w_1}\tilde{\psi}_{w_1} & \\ +\frac{\sigma}{2}\{(\psi^*_{w_1} - \psi^o_{w_1})^T(\psi^*_{w_1} - \psi^o_{w_1})\} \quad \leq \quad -\frac{b}{2}s^2 - \frac{b}{2\gamma}\tilde{\theta}^T_{w_1}\tilde{\theta}_{w_1} - \frac{b}{2\gamma}\tilde{\psi}_{w_1}\tilde{\psi}_{w_1} + \lambda & \end{aligned} \quad (27)$$

where we define

$$b \leq \min\{2k, \sigma\gamma\} \quad (28)$$

$$\lambda = \kappa\epsilon\bar{\psi}^*_{w_1} + \frac{\sigma}{2}\{(\theta^*_{w_1} - \theta^o_{w_1})^T(\theta^*_{w_1} - \theta^o_{w_1}) + (\psi^*_{w_1} - \psi^o_{w_1})(\psi^*_{w_1} - \psi^o_{w_1})\}. \quad (29)$$

From (26), we can see that $\dot{V}$ is not negative definite (nor negative semi-definite). Therefore asymptotic convergence cannot be concluded. However, for bounded λ, the solution, V, can be shown to be uniformly ultimately bounded (u.u.b).

The solution to inequality (26) satisfies

$$\begin{aligned}V \leq & V_o e^{-bt} + \lambda\int_0^t e^{-b(t-\tau)}d\tau \\ = & e^{-bt}\{V_o - \frac{\lambda}{b}\} + \frac{\lambda}{b}. \end{aligned} \quad (30)$$

Therefore, using equation (16) and (30) we obtain that for any $\rho \geq \sqrt{\frac{2\lambda}{b}}$ there exits a T such that for all $t \geq T$, the variable s in the Lyapunov equation satisfies

$$|s(t)| \leq \rho. \quad (31)$$

This implies that the components of V are also u.u.b. The bound given by (31) also implies that the tracking error is bounded as shown below. Define $\omega(t) = \int_0^t e(\tau)d\tau$. Using this we can write (25) as

$$\dot{\omega} = -c\,\omega + s. \quad (32)$$

Since equation (32) represents a linear system driven by a bounded input, s, we know that the state, e, is bounded also. Thus, we have uniformly ultimately bounded tracking error.

We now address the well-definedness of the control by restricting the adaptation law of the parameter which multiplies the control signal. In order to insure that the control $u(t)$ remains bounded, the denominator in equation (14) must be bounded away from zero. We insure this by forcing θ_{w_2} such that $\theta_{w_2}^T \xi > |\psi_{w_2}^*|$. Since we know that the approximation of w_2 is bounded from below by $\underline{w}_2(x)$, we require $\underline{w}_2 > |\delta_{w_2}|$. Therefore, to insure bounded control we impose the limits on the approximation, turning off the update of the parameter vector elements if the update would drive them below $\underline{\theta}_{w_{2j}}$ where j refers to the j^{th} element of the parameter vector. Since we have also assumed an upper bound on the *true* functions, we turn off the parameter updates if the update would drive them above $\overline{\theta}_{w_{2j}}$. Since this will leave additional terms in the Lyapunov derivative, we must consider the stability impact.

To impose these bounds on the parameter elements, we now modify the parameter rule given in equation (12). Each element of the parameter vector is now updated according to

$$\dot{\theta}_{w_{2j}} = \begin{cases} 0 & \text{if } \{\theta_{w_{2j}} \leq \underline{\theta}_{w_{2j}} \text{ and } \gamma s u \xi_j \leq 0\} \\ 0 & \text{if } \{\theta_{w_{2j}} \geq \overline{\theta}_{w_{2j}} \text{ and } \gamma s u \xi_j \geq 0\} \\ \gamma s u \xi_j & \text{otherwise.} \end{cases} \tag{33}$$

Elements of the parameter vector, $\theta_{w_{2j}}$ that are updated using the nominal parameter update equation, (12), will not leave residual terms in the Lyapunov derivative. However, those elements of the parameter vector that have their update zeroed, leave terms in the Lyapunov derivative that must be considered.

Let us designate the elements in the parameter vector that would violate their upper bound if updated according to (12) as $\theta_{w_{2i}}$. Let us assume that these parameter elements are q in number. Similarly, we define those parameter elements that would violate their lower bounds by using (12) as $\theta_{w_{2j}}$ and assume that these are r in number. Thus, the terms that would be added to the Lyapunov derivative due to the zeroing of the parameter updates would have the form

$$-\sum_{i=1}^{q} \tilde{\theta}_{w_{2_i}} s u \xi_i - \sum_{j=1}^{r} \tilde{\theta}_{w_{2_j}} s u \xi_j. \tag{34}$$

The first term of (34) can be written as

$$-\sum_{i=1}^{q} (\theta_{w_{2i}} - \theta_{w_{2i}}^*) s u \xi_i. \tag{35}$$

Since we know that the $\theta_{w_{2_i}}$ is at the upper limit of the i^{th} element of the parameter, $\bar{\theta}_{w_{2_i}}$, and the optimal value, $\theta_{w_{2_i}}^*$, is below the maximum value; we can conclude that $(\theta_{w_{2^i}} - \theta_{w_{2i}}^*) \geq 0$. Also,

since we know that the nominal update for $\theta_{w_{2_i}}$ is increasing, $\gamma > 0$, and because $\xi(x) > 0 \ \forall x$, then we know that $us > 0$. Thus, we know that the first term of (34) is negative, and therefore, does not de-stabilize the system.

The second term of (34) can be written as

$$-\sum_{j=1}^{r}(\theta_{w_{2_j}} - \theta^*_{w_{2_j}})su\xi_j. \tag{36}$$

Here we know that the $\theta_{w_{2_j}}$ is at the lower limit of the i^{th} element of the parameter, $\underline{\theta}_{w_{2_j}}$, and the optimal value, $\theta^*_{w_{2_j}}$, is above the minimum value; we can conclude that $(\theta_{w_{2_j}} - \theta^*_{w_{2_j}}) \leq 0$. Also, since we know that the nominal update for $\theta_{w_{2_j}}$ is decreasing, $\gamma > 0$, and because $\xi(x) > 0 \ \forall x$, then we know that $us < 0$. Thus, we know that the second term of (34) is negative and does not de-stabilize the system either. Therefore, the system tracking errors and parameter errors are guaranteed to be uniformly ultimately bounded and the control is bounded for all x values in the domain, Ω.

The problem we have described is tracking control for a SISO, affinely represented model with parametric uncertainty due to unknown $\theta^*_{w_1}$ and $\theta^*_{w_2}$, bounding uncertainty due to the unknown reconstruction error bounds, $\psi^*_{w_1}$, and unknown but bounded w_2. The neural networks approximate the unknown system nonlinearities w_i, the vector sliding mode control with adaptive bounding insures boundedness, and the integral action improves transient response.

3 SISO Simulation Example

To illustrate the neural control system design, a SISO model has been defined in the form of equation (1) which is affine in the controls and has nonlinear characteristics. The model doesn't represent a physical system, but it does possess an uncertain state dependent nonlinear quantity, $w_1(x)$, and an uncertain control multiplier, w_2, where

$$w_1(x) = 2 + 0.5\,x + 0.2500\,x^2 - 0.0104\,x^3 - 0.0078\,x^4 \tag{37}$$

$$w_2 = 5. \tag{38}$$

We define the uncertainty used to bound w_2 as Δ_2 where

$$\Delta_2 = 0.5. \tag{39}$$

Further, we define the domain of the state, x, as $\Omega = [-5, 5]$.

The desired trajectory, x_d, is given by a combination of sine waves of various frequencies, square waves and a saw tooth function. This trajectory provides abundant excitation for the function approximation in some parts of the state space.

The function approximation is performed by a linear-in-the-parameters neural network of the form given in equation (2). The basis functions, $\xi(x)$, are Radial Basis Functions (RBFs) consisting of 51 evenly spaced functions over a domain of $[-5, 5]$ with standard deviation of 0.15.

Simulation results have been generated for the SISO plant defined above, x_d, and the control structure given in Section 2. To illustrate impact of learning on closed loop system performance, the neural networks were initialized with weights that resulted in $\hat{w}_1(x)$ that was about half of the *true* function and $\hat{w}_2(x) \approx \underline{w}_2(x)$. Closed loop tracking results are shown in Figure 1 using three separate five second time slices of one time history. These plots show how the tracking performance

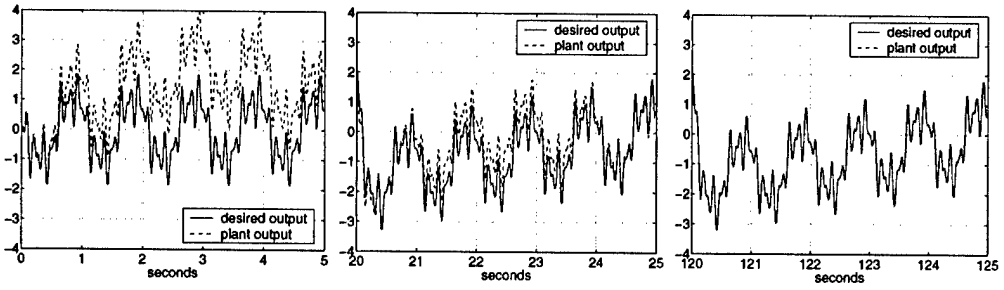

Figure 1: Tracking Performance

improves as time progresses. The top plot shows sluggish performance during the start of the simulation where the function approximation is relatively poor. The middle plot shows that the performance is improved during the interval between 20 and 25 seconds due to improved function approximation. The bottom plot show the tracking between 120 and 125 seconds where, at that point in the simulation, the function approximation is very good and the traces essentially lie on top of each other. Thus, as time progresses, the control system learns to improve its tracking performance.

During the 125 seconds of simulation the weights of the neural network function approximators adjust. This can be seen in Figure 2, which show the function approximation error for w_1 (left) and w_2 (right). At the beginning of the simulation the weights associated with $\hat{w}_1$ were set to values that made the approximation about half of the true value of the function throughout the entire x domain. To provide a valid basis for evaluation of the control system's function approximation, the best possible approximation for w_1 was found off-line using a gradient algorithm and random samples from the domain until the error grew very small. Thus, in Figure 2 we see the starting point for the simulation, the best possible approximation, and the approximation that results after 125 seconds of simulation. It is clear that the approximation is greatly improved after the simulation over some of the domain. It should not be surprising that the fit is nearly identical to the initial approximation in regions of the domain that the state did not reach. Because of the local nature of

the basis functions used, the approximation will only train in regions that are visited by the state.

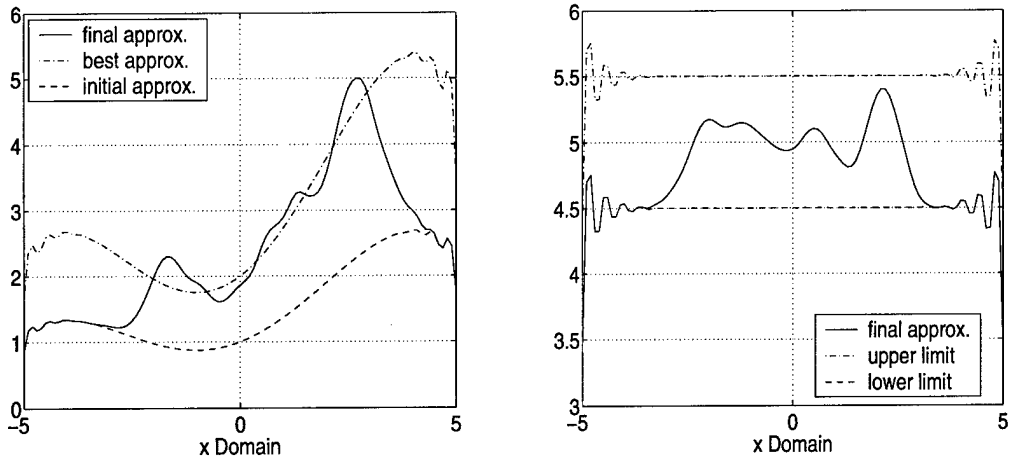

Figure 2: Function approximation of w_1 (on the left) and w_2 (on the right).

In the case of $\hat{w}_2$, uncertainty bounds are imposed on the weights as shown in equation (33). This causes the approximation of the network to be constrained to lie within upper and lower bounding limits. The uncertainty bound given by equation (39) implies that the value of w_2 should remain between 4.5 and 5.5 for all values of x. However, to place bounds on the approximation here we use the *best* approximation of the upper and lower bounds to provide upper and lower bounds for θ_{w_2}.

Figure 2 (right) shows the results of the w_2 approximation. The best approximation of the upper and lower bounds are shown as dashed and dash-dot lines while the approximation after the simulation is complete is shown by the solid line. The simulation is initialized with the $\theta_{w_2} = \underline{\theta}_{w_2}$. For this reason, approximation at the end of the simulation is nearly identical to the *best* approximation of the lower bound in regions of the domain not visited.

It is clear from Figure 2 that the approximations improve greatly during the course of the simulation. This is largely the cause of the improved tracking shown in Figure 1.

4 Two-Input Two-Output Control Design

In many systems, including control of high performance aircraft systems, it is common to have multiple control inputs affecting the state variables. One of the approaches for handling such systems is to use decoupling methods for isolating the effect of a control input on a measured output. However, such methods require that the designer know a priori the dynamics of the system such that coupling between control inputs can be cancelled (or minimized). The problem becomes considerably

more difficult if the system is nonlinear and some of these nonlinearities are unknown. A key challenge is to handle unknown nonlinearities multiplying the control inputs. In this section, we design a neural control scheme for a two-input two-output system and analyze the stability properties of the closed-loop system.

We formulate the two-input two-output (TITO) problem as an extension of the SISO problem and consider the problem of tracking control for a TITO plant with full state measurement. The state vector is defined as X, where $X = [x\ z]^T$, where x and z are the states, $(x, z) \in \Omega \subset \mathbb{R}^2$, and Ω is a domain of interest. The control vector is defined as U, where $U = [u\ v]^T$ and u and v are the control inputs. The plant dynamics for the system are written as

$$\dot{X} = W_A + W_B U \tag{40}$$

where W_A is a vector of functions, W_B is a matrix of functions and they are given by

$$W_A = \begin{bmatrix} w_1(x,z) \\ w_2(x,z) \end{bmatrix} \quad \text{and} \quad W_B = \begin{bmatrix} w_3(x,z) & w_4(x,z) \\ w_5(x,z) & w_6(x,z) \end{bmatrix}. \tag{41}$$

The functions, $w_i(x, z)$ for $i = 1, \ldots, 6$, are assumed to be unknown and the system represented by equation (40) is assumed to be stabilizable. The control system objective is to use the control inputs signals, u and v, to make the states, x and z, track externally provided desired reference trajectories, x_d and z_d, respectively.

The unknown functions and their associated uncertainties are defined for the TITO system in a manner similar to what was used for the SISO system. Each of the unknown functions is modeled using a linearly parameterized combination of basis functions:

$$\hat{w}_i(x, z) = \theta_{w_i}^T \xi(x, z) \quad \text{for } i = 1, \ldots, 6 \tag{42}$$

where $\xi(x, z)$ is the $m \times 1$ vector of basis functions and θ_{w_i} are the $m \times 1$ parameter vectors, similar to those defined in equation (2) for the SISO formulation. Just as the true functions, w_i, were used to define vector and matrix quantities in (41), function approximations $\hat{w}_i$ are used to define $\hat{W}_A$ and $\hat{W}_B$.

The *best* parameter vector and *best* approximation for each unknown function are defined according to equations (43) and (44) respectively.

$$\theta_{w_i}^* = \arg \min_{\theta_{w_i} \in \mathbb{R}^m} \{ \sup_{(x,z)\in\Omega} |w_i(x, z) - \theta_{w_i}^T \xi(x, z)| \} \tag{43}$$

$$w_i^*(x, z) = \theta_{w_i}^{*T} \xi(x, z) \tag{44}$$

The reconstruction errors are given by

$$\delta_{w_i}(x, z) = w_i(x, z) - w_i^*(x, z) \tag{45}$$

and for w_1 and w_2 we define the reconstruction error bounds on equation (45) as $\psi^*_{w_i}$ where

$$\psi^*_{w_i} = \sup_{(x,z)\in\Omega} |\delta_{w_i}(x,z)|. \tag{46}$$

We further define a bound, $\bar{\psi}^*_{w_i} = \max\{\psi^*_{w_i}, \psi^o_{w_i}\}$, where $\psi^o_{w_i}$ is a design parameter.

In the following neural control design, the reconstruction error is addressed using an adaptive bound for w_1 and w_2. However, the reconstruction errors in the approximations of the functions which multiply the control inputs are dealt with differently. These reconstruction errors, δ_i for $i = 3, \ldots, 6$, are defined as

$$w_3(x,z)u = \theta^{*T}_3\xi(x)u + \delta_3(x,z,u) \tag{47}$$

$$w_4(x,z)v = \theta^{*T}_4\xi(x)v + \delta_4(x,z,v) \tag{48}$$

$$w_5(x,z)u = \theta^{*T}_5\xi(x)u + \delta_5(x,z,u) \tag{49}$$

$$w_6(x,z)v = \theta^{*T}_6\xi(x)v + \delta_6(x,z,v). \tag{50}$$

For these functions of the W_B matrix, the reconstruction error bounds are not adaptive and are defined as $\bar{\delta}_{w_i} = \max|\delta_{w_i}|$ for $i = 3, \ldots, 6$ and where δ_{w_i} is given by equation (45).

We further assume that the functions, $w_{3\ldots6}$, are within some known bounds. These bounds are defined as $\underline{w}_i(x,z)$ and $\bar{w}_i(x,z)$ where

$$\underline{w}_i(x,z) < w_i(x,z) < \bar{w}_i(x,z). \tag{51}$$

and where $\bar{w}_i(x,z) = \bar{\theta}^T_{w_i}\xi(x,z)$ and $\underline{w}_i(x,z) = \underline{\theta}^T_{w_i}\xi(x,z)$. Therefore, we impose the restriction on the function estimates by preventing the associated parameter vectors from exceeding the upper and lower parameter vectors, $\bar{\theta}_{w_i}$ and $\underline{\theta}_{w_i}$.

We define the tracking errors for the x output, $e_x = x - x_d$, and for the z output, $e_z = z - z_d$. We then use these errors to define sliding mode variables for x and z and write this as a vector of components

$$s = \begin{bmatrix} s_x \\ s_z \end{bmatrix} = \begin{bmatrix} e_x + c_1 \int e_x d\tau \\ e_z + c_2 \int e_z d\tau \end{bmatrix} \tag{52}$$

where c_1 and c_2 are positive design constants. Using these sliding mode quantities in the control law induces proportional plus integral control action [17, 31].

The rules for updating the parameter vectors are defined similar to what was shown in Section 2 and are given by

$$\dot{\theta}_{w_1} = \gamma\{\xi s_x - \sigma(\theta_{w_1} - \theta^o_{w_1})\} \tag{53}$$

$$\dot{\theta}_{w_2} = \gamma\{\xi s_z - \sigma(\theta_{w_2} - \theta^o_{w_2})\} \tag{54}$$

and

$$\dot{\theta}_{w_3} = \gamma \xi s_x u \tag{55}$$

$$\dot{\theta}_{w_4} = \gamma \xi s_x v \tag{56}$$

$$\dot{\theta}_{w_5} = \gamma \xi s_z u \tag{57}$$

$$\dot{\theta}_{w_6} = \gamma \xi s_z v \tag{58}$$

where σ is a leakage constant and $\theta^o_{w_1}$ and $\theta^o_{w_2}$ are design parameters. The update rules for the adaptive bounds are given by

$$\dot{\psi}_{w_1} = \gamma\{s_x \tanh(\frac{s_x}{\epsilon}) - \sigma(\psi_{w_1} - \psi^o_{w_1})\} \tag{59}$$

$$\dot{\psi}_{w_2} = \gamma\{s_z \tanh(\frac{s_z}{\epsilon}) - \sigma(\psi_{w_2} - \psi^o_{w_2})\} \tag{60}$$

where $\psi^o_{w_1}$ and $\psi^o_{w_2}$ are also design parameters.

We now write the vector of control inputs as U where

$$U = \hat{W}_B^{-1} \begin{bmatrix} -\theta_{w_1}\xi - \psi_{w_1}\tanh(\frac{s_x}{\epsilon}) + \dot{x}_d - c_1 e_x - k s_x - (\bar{\delta}_{w_3} + \bar{\delta}_{w_4})\mathrm{sgn}(s_x) \\ -\theta_{w_2}\xi - \psi_{w_2}\tanh(\frac{s_z}{\epsilon}) + \dot{z}_d - c_2 e_z - k s_z - (\bar{\delta}_{w_5} + \bar{\delta}_{w_6})\mathrm{sgn}(s_z) \end{bmatrix} \tag{61}$$

where k is a positive real number.

We insure the existence of the controls, U, by making restrictions that guarantee that $\hat{W}_B^{-1}$ is defined. We require that the sign of the determinant of $\hat{W}_B$ remains constant over the range of possible variations of the function approximation and we also require that each of the elements of $\hat{W}_B$ are sign definite. We assume that each of these elements are positive and

$$\underline{w}_3 \underline{w}_6 - \bar{w}_4 \bar{w}_5 > 0 \ \forall (x, z) \in \Omega \tag{62}$$

insuring that $\hat{W}_B$ has a positive determinant.

In cases where the sign of the determinant is negative or the sign of some or all of the elements of $\hat{W}_B$ are negative, restrictions similar to (62) can be made so long as the elements are sign definite.

Stability of the system is addressed by using Lyapunov analysis. For the TITO case, consider the Lyapunov function given by

$$V = \frac{1}{2} s^T s + \sum_{i=1}^{6} \frac{\tilde{\theta}^T_{w_i} \tilde{\theta}_{w_i}}{2\gamma} + \sum_{i=1}^{2} \frac{\tilde{\psi}^2_{w_i}}{2\gamma} \tag{63}$$

where $\gamma > 0$ is the adaptive gain, $\tilde{\theta}_{w_i} = \theta_{w_i} - \theta^*_{w_i}$ for $i = 1...6$, and $\tilde{\psi}_{w_i} = \psi_{w_i} - \bar{\psi}^*_{w_i}$ for $i = 1, 2$.

Taking the derivative of equation (63) and using equation (52), we get

$$\dot{V} = s^T \begin{bmatrix} c_1 e_x - \dot{x}_d \\ c_2 e_z - \dot{z}_d \end{bmatrix} + s^T\{W_A + W_B U\} + \sum_{i=1}^{6} \frac{\tilde{\theta}^T_{w_i} \dot{\theta}_{w_i}}{\gamma} + \sum_{i=1,2} \frac{\tilde{\psi}_{w_i} \dot{\psi}_{w_i}}{\gamma}. \tag{64}$$

If we now add and subtract $s^T\hat{W}_A$ and $s^T\hat{W}_BU$ in equation (64) we obtain

$$\begin{aligned}\dot{V} &= s^T\{\begin{bmatrix} c_1e_x - \dot{x}_d \\ c_2e_z - \dot{z}_d \end{bmatrix} + \begin{bmatrix} \delta_{w_1} \\ \delta_{w_2} \end{bmatrix} - \begin{bmatrix} \tilde{\theta}_{w_1}^T \\ \tilde{\theta}_{w_2}^T \end{bmatrix}\xi + \begin{bmatrix} \delta_{w_3} & \delta_{w_4} \\ \delta_{w_5} & \delta_{w_6} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &- \begin{bmatrix} \tilde{\theta}_{w_3}^T\xi & \tilde{\theta}_{w_4}^T\xi \\ \tilde{\theta}_{w_5}^T\xi & \tilde{\theta}_{w_6}^T\xi \end{bmatrix}U + \hat{W}_A + \hat{W}_BU\} + \sum_{i=1}^{6}\frac{\tilde{\theta}_{w_i}^T\dot{\theta}_{w_i}}{\gamma} + \sum_{i=1,2}\frac{\tilde{\psi}_{w_i}\dot{\psi}_{w_i}}{\gamma}. \end{aligned} \tag{65}$$

Using the reconstruction error bounds we can write equation (65) as

$$\begin{aligned}\dot{V} &\leq s^T\{\begin{bmatrix} c_1e_x - \dot{x}_d \\ c_2e_z - \dot{z}_d \end{bmatrix} - \begin{bmatrix} \tilde{\theta}_{w_1}^T \\ \tilde{\theta}_{w_2}^T \end{bmatrix}\xi + \begin{bmatrix} \delta_{w_3} & \delta_{w_4} \\ \delta_{w_5} & \delta_{w_6} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \tilde{\theta}_{w_3}^T\xi & \tilde{\theta}_{w_4}^T\xi \\ \tilde{\theta}_{w_5}^T\xi & \tilde{\theta}_{w_6}^T\xi \end{bmatrix}U \\ &+ \hat{W}_A + \hat{W}_BU\} + |s|^T\begin{bmatrix} \bar{\psi}_{w_1}^* \\ \bar{\psi}_{w_2}^* \end{bmatrix} + \sum_{i=1}^{6}\frac{\tilde{\theta}_{w_i}^T\dot{\theta}_{w_i}}{\gamma} + \sum_{i=1,2}\frac{\tilde{\psi}_{w_i}\dot{\psi}_{w_i}}{\gamma} \end{aligned} \tag{66}$$

where $|s| = [|s_x| \;\; |s_z|]^T$.

We now take the parameter update rules shown in equations (53-58) and plug them into (66) to obtain

$$\begin{aligned}\dot{V} &\leq s^T\{\begin{bmatrix} c_1e_x - \dot{x}_d \\ c_2e_z - \dot{z}_d \end{bmatrix} + \begin{bmatrix} \delta_{w_3} & \delta_{w_4} \\ \delta_{w_5} & \delta_{w_6} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \hat{W}_A + \hat{W}_BU\} \\ &+ |s|^T\begin{bmatrix} \bar{\psi}_{w_1}^* \\ \bar{\psi}_{w_2}^* \end{bmatrix} - \begin{bmatrix} \sigma & \sigma \end{bmatrix}\begin{bmatrix} \tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) \\ \tilde{\theta}_{w_2}^T(\theta_{w_2} - \theta_{w_2}^o) \end{bmatrix} + \sum_{i=1,2}\frac{\tilde{\psi}_{w_i}\dot{\psi}_{w_i}}{\gamma}. \end{aligned} \tag{67}$$

The control from equation (61) is substituted into (67) to obtain

$$\begin{aligned}\dot{V} &\leq |s|^T\begin{bmatrix} \bar{\psi}_{w_1}^* \\ \bar{\psi}_{w_2}^* \end{bmatrix} - s^T\begin{bmatrix} \{\psi_{w_1}\tanh(\frac{s_x}{\epsilon}) + ks_x\} \\ \{\psi_{w_2}\tanh(\frac{s_z}{\epsilon}) + ks_z\} \end{bmatrix} - \begin{bmatrix} \sigma & \sigma \end{bmatrix}\begin{bmatrix} \tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) \\ \tilde{\theta}_{w_2}^T(\theta_{w_2} - \theta_{w_2}^o) \end{bmatrix} \\ &+ \sum_{i=1,2}\frac{\tilde{\psi}_{w_i}\dot{\psi}_{w_i}}{\gamma}. \end{aligned} \tag{68}$$

Using (20), (59) and (60) we can write (68) as

$$\dot{V} \leq -ks^Ts + \bar{\psi}_{w_1}^*\kappa\epsilon + \bar{\psi}_{w_2}^*\kappa\epsilon - \begin{bmatrix} \sigma & \sigma \end{bmatrix}\begin{bmatrix} \tilde{\theta}_{w_1}^T(\theta_{w_1} - \theta_{w_1}^o) + \tilde{\psi}_{w_1}^T(\psi_{w_1} - \psi_{w_1}^o) \\ \tilde{\theta}_{w_2}^T(\theta_{w_2} - \theta_{w_2}^o) + \tilde{\psi}_{w_2}^T(\psi_{w_2} - \psi_{w_2}^o) \end{bmatrix} \tag{69}$$

which can also be written as

$$\dot{V} \leq -ks_x^2 - ks_z^2 + \bar{\psi}_{w_1}^*\kappa\epsilon + \bar{\psi}_{w_2}^*\kappa\epsilon - \sigma\sum_{i=1}^{2}\{\tilde{\theta}_{w_i}(\theta_{w_i} - \theta_{w_i}^o) + \tilde{\psi}_{w_i}(\psi_{w_i} - \psi_{w_i}^o)\}. \tag{70}$$

As was done in Section 2, we can show stability [23] where $\dot{V} \leq -\alpha V + \lambda$, $\alpha = \min(2k, \sigma\gamma)$ and

$$\lambda = \kappa\epsilon(\bar{\psi}_{w_1}^* + \bar{\psi}_{w_2}^*) + \frac{\sigma}{2}\sum_{i=1}^{2}\{(\theta_{w_i}^* - \theta_{w_i}^o)^T(\theta_{w_i}^* - \theta_{w_i}^o)\} + (\psi_{w_1}^* - \psi_{w_1}^o)^2 + (\psi_{w_2}^* - \psi_{w_2}^o)^2. \tag{71}$$

In the case that the parameter update equations (55-58) attempt to drive the parameter values of $\{\theta_{w_i} \mid i = 3, \ldots, 6\}$ beyond their bounds, we zero the update to the parameter. The terms left remaining in the Lyapunov derivative due to this zeroing are similar to those shown in (34). Correspondingly, the stability analysis that was done for the SISO case shown in Section 2 can be used to show zeroing parameter updates does not de-stabilize the system in the TITO case either. Therefore, we obtain uniform ultimate boundedness of the system tracking errors, the parameters and the adaptive bounds.

5 TITO Simulation Example

To illustrate the neural control system design, a model has been defined which is affine in the controls and has nonlinear characteristics.

The model is given by

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} w_{11}(x) + w_{12}(z) \\ w_{21}(x) + w_{22}(z) \end{bmatrix} + \begin{bmatrix} w_3 & w_4(x) \\ w_5(x) & w_6 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{72}$$

where

$$w_{11}(x) = 2 + 0.5\,x + 0.2500\,x^2 - 0.0104\,x^3 - 0.0078\,x^4 \tag{73}$$

$$w_{12}(z) = z \tag{74}$$

$$w_{21}(x) = 5 + x - 0.4821\,x^2 - 0.0357\,x^3 + 0.0179\,x^4 \tag{75}$$

$$w_{22}(z) = -1.7z \tag{76}$$

$$w_3 = w_6 = 3 \tag{77}$$

$$w_4(x) = w_5(x) = x/8 + 1 \tag{78}$$

and we define the domain of the plant state as $\Omega = \{(x, z) \mid x \in [-5, 5],\ z \in [-5, 5]\}$.

This model differs slightly from the model given by equations (40) and (41). In (72) we allow the $w_1(x, z)$ and $w_2(x, z)$ terms to be written as separate functions of x and z. We also write w_4 and w_5 as functions of x only, while w_3 and w_6 are constants. These changes simplify the function approximations for presentation purposes.

The same function approximation structure is used here for the TITO example as was used for the SISO example. Namely, a linear-in-the-parameters neural network of the form given in equation (2). For the TITO case, each set of Radial Basis Functions (RBFs), $\xi(x)$ and $\xi(z)$, consists of 51 evenly spaced functions over a domain of $[-5, 5]$ with standard deviation of 0.1.

The desired trajectories, x_d and z_d, are given by a combination of sine waves of various frequencies and square waves. As in the SISO simulation, these trajectories provide abundant excitation for the

function approximation. The magnitudes of x_d and z_d are also defined to allow the state trajectories to remain within Ω.

Simulation results have been generated for the TITO plant defined above, x_d and z_d, and the control structure given in Section 4. To illustrate the impact of learning on closed loop system performance, the neural networks were initialized with weight values that resulted in poor initial function approximations. Initial θ_{w_1} and θ_{w_2} parameter vectors were chosen such that $\hat{w}_1 \approx 0$ and $\hat{w}_2 \approx 0$ throughout Ω. The weights for $\hat{w}_i$ where $i = 3, \ldots, 6$ are initialized to the worst case bounds, e.g., those which result in the minimum determinant of W_B. Thus, the $\hat{w}_3$ and $\hat{w}_6$ approximators are initialized with $\underline{\theta}_{w_3}$ and $\underline{\theta}_{w_6}$ and $\hat{w}_4$ and $\hat{w}_5$ are initialized with $\bar{\theta}_{w_4}$ and $\bar{\theta}_{w_6}$. Successful system performance was to be demonstrated by tracking performance improvements through time as the function approximators more accurately mapped the system dynamics.

Closed loop tracking results are shown in the time histories in Figures 3. The traces on the left of Figure 3 show rather poor tracking for the first five seconds of the simulation. This is due to the fact that, initially, the function approximation outputs do not accurately represent the dynamics. However, after 25 seconds of simulation, tracking is substantially improved as can be seen on the right two plots. The improvement in the tracking of the closed-loop system is due to the improved function approximation that results as the simulation runs.

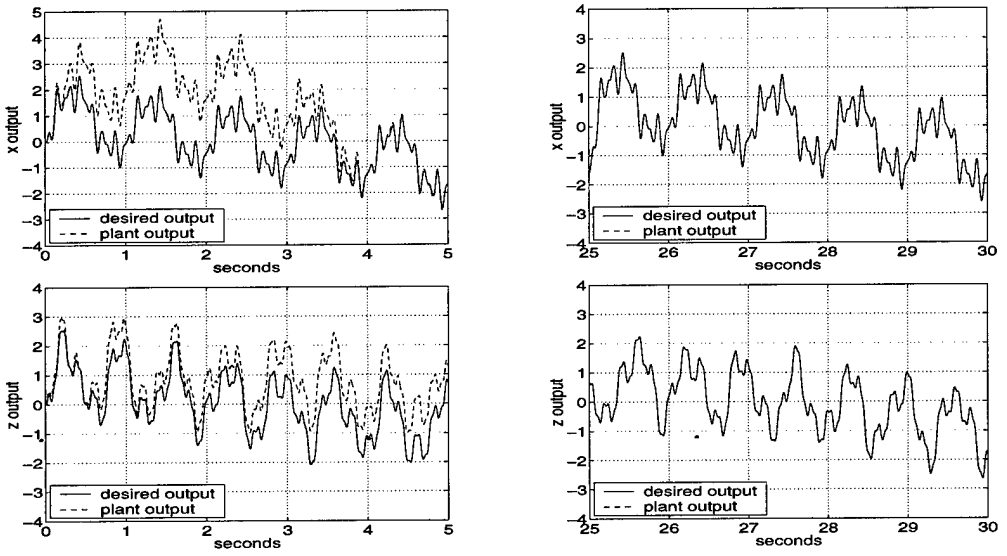

Figure 3: On the left the initial five seconds of states x(t) and z(t) tracking the desired trajectories x_d and z_d; on the right, the tracking performance is shown after 25 seconds.

The improvement in function approximation can be seen in Figures 4-5. The approximations

for w_1 and w_2 are plotted along the line $x = z$ in the (x, z) plane in Figure 4. This provides some indication of fit over the entire domain without the complexity of surface plots with multiple surfaces. Figure 4 shows the function approximation after the simulation is complete, the initial approximation, and the best possible approximation given the structure of the approximator. It should be noted that although the approximations improve greatly in those regions of the (x, z) domain visited during the simulation, the approximation remains very near the initial value in regions not visited during the simulation. This is due to local nature of the RBFs.

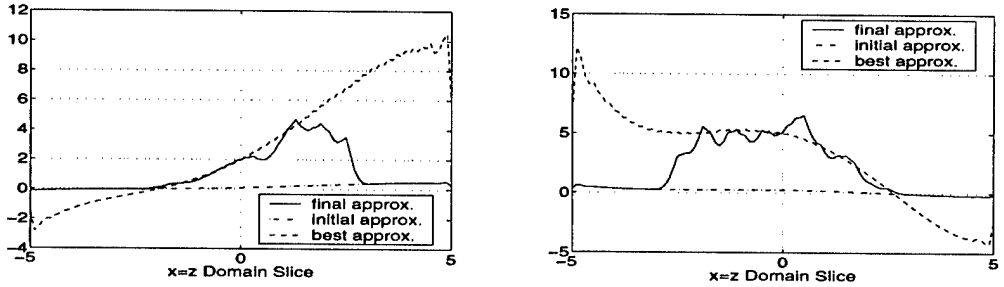

Figure 4: Function approximation of $w_1(x, z)$ (left) and $w_2(x, z)$ (right).

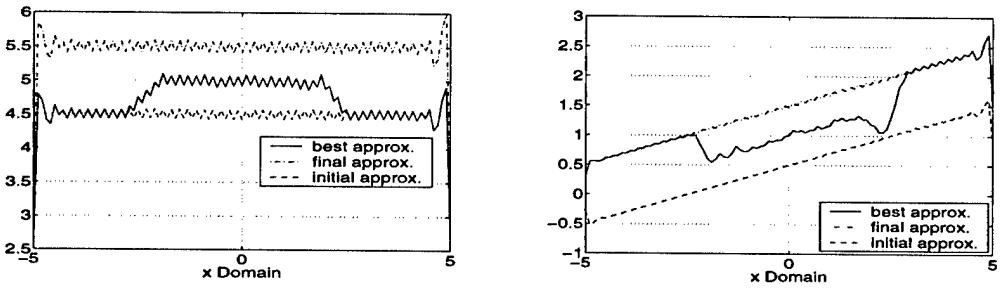

Figure 5: Function approximation w_3 and w_6 (left) and $w_4(x)$ and $w_5(x)$ (right).

Figure 5 shows the approximation of the functions which multiply the control inputs. Since these functions depend only on x, these plots have x as the horizontal axis and because $w_3 = w_6$ and $w_4 = w_5$, only two plots are shown. These plots show the *best approximation* to the upper bounds and lower bounds, and the approximation after the simulation has run for 30 seconds. The plots in Figure 5 shows good approximation of the *true* function value in parts of Ω which are visited during the simulation and remain near the boundaries in other regions of Ω.

6 Application to MIMO systems

To this point we have only addressed SISO and TITO systems. However, the approach that has been applied to the TITO problem is applicable to MIMO systems with full state output where the number of inputs is equal to the outputs. For such systems the *equations of motion* written in equation (40) can be written in a more general form as

$$\begin{aligned}
\dot{x}_1 &= w_{1,1} + \sum_{i=1}^{n} w_{1,i+1} u_i^* \\
\dot{x}_2 &= w_{2,1} + \sum_{i=1}^{n} w_{2,i+1} u_i^* \\
&\vdots \qquad\qquad\qquad \vdots \\
\dot{x}_n &= w_{n,1} + \sum_{i=1}^{n} w_{n,i+1} u_i^*
\end{aligned} \tag{79}$$

where $w_{i,j}$ are functions of the states $(x_1, x_2, ...x_n)$ and basis functions are therefore functions of all states. Functions would be approximated by $\hat{w}_{i,j}(x_1, x_2, ...x_n)$ where

$$\hat{w}_{i,j}(x_1, x_2, ...x_n) = \theta_{i,j}^T \, \xi(x_1, x_2, ...x_n). \tag{80}$$

An adaptive bound would be used for $\hat{w}_{i,1}$ and fixed bounds would be used for those functions which multiply the controls. The update rules would be extensions of equations (53-60), and the control, $U = \hat{W}_B^{-1} L$, would be an n-dimensional extension of equation (61), where $\hat{W}_B$ would become an n-by-n matrix given by

$$\hat{W}_B = \begin{bmatrix} w_{1,2}(x_1, ...x_n) & w_{1,3}(x_1, ...x_n) & \cdots & w_{1,n+1}(x_1, ...x_n) \\ \vdots & \ddots & \vdots & \\ w_{n,2}(x_1, ...x_n) & w_{n,3}(x_1, ...x_n) & \cdots & w_{n,n+1}(x_1, ...x_n) \end{bmatrix}, \tag{81}$$

and where the matrix, L, is given by

$$L = \begin{bmatrix} -\theta_{w_{1,1}}\xi - \psi_{w_{1,1}} \tanh(\frac{s_{x_1}}{\epsilon}) + \dot{x}_{1_d} - c_1 e_{x_1} - k s_{x_1} - (\bar{\delta}_{w_{1,2}} + \cdots + \bar{\delta}_{w_{1,n+1}}) \mathrm{sgn}(s_{x_1}) \\ -\theta_{w_{2,1}}\xi - \psi_{w_{2,1}} \tanh(\frac{s_{x_2}}{\epsilon}) + \dot{x}_{2_d} - c_1 e_{x_2} - k s_{x_2} - (\bar{\delta}_{w_{2,2}} + \cdots + \bar{\delta}_{w_{2,n+1}}) \mathrm{sgn}(s_{x_2}) \\ \vdots \\ -\theta_{w_{n,1}}\xi - \psi_{w_{n,1}} \tanh(\frac{s_{x_n}}{\epsilon}) + \dot{x}_{n_d} - c_1 e_{x_n} - k s_{x_n} - (\bar{\delta}_{w_{n,2}} + \cdots + \bar{\delta}_{w_{n,n+1}}) \mathrm{sgn}(s_{x_n}) \end{bmatrix}. \tag{82}$$

Since the control solution given above includes $\hat{W}_B^{-1}$, we must insure that the determinant of $\hat{W}_B$ maintains one sign. Therefore, this matrix must be checked throughout the ranges of variations to insure that the sign of the determinant does not change. This approach for the MIMO case results in uniformly ultimately bounded stability just as was shown in the SISO and TITO cases.

7 Conclusion

A neural control approach for a class of nonlinear systems has been developed. Lyapunov analysis has been used to insure bounded tracking errors and robust adaptive methods prevent parameter drift. Results have been illustrated with Single-Input Single-Output and Two-Input Two-Output plants. Also, extensions to more general Multiple-Input Multiple-Output plants has been supported with analysis.

References

[1] B. R. Barmish, M. Corless, and G. Leitmann. A new class of stabilizing controllers for uncertain dynamical systems. *SIAM Journal on Control and Optimization*, 21:246–255, 1983.

[2] A. Calise and R. Rysdyk. Nonlinear adaptive flight control using neural networks. *IEEE Contr. Sys. Mag.*, 18(6):14–25, December 1998.

[3] P. Chandler, M. Mears, and M. Pachter. On-line optimizing networks for reconfigurable control. In *Proceedings of the 32nd Conference on Decision and Control*, volume 3, pages 2272–2277, 1993.

[4] F.-C. Chen and H. K. Khalil. Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Trans. on Autom. Control*, 40:791–801, May 1995.

[5] M. Corless and G. Leitmann. Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain systems. *IEEE Trans. on Autom. Control*, 26:1139–1144, 1981.

[6] S. Fabri and V. Kadirkamanathan. Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Trans. on Neural Net.*, 7:1151–1167, September 1996.

[7] J. Farrell. Stability and approximator convergence in nonparametric nonlinear adaptive control. *IEEE Trans. on Neural Net.*, 9:1008–1020, September 1998.

[8] J. Franklin. Historical perspective and state of the art in connectionist learning control. In *Proceedings of the 28th Conference on Decision and Control*, pages 1730–1736, 1989.

[9] D. Godbole. Control and coordination of uninhabited combat air vehicles. In *Proceedings of the 1999 American Control Conference*, pages 1487–1490, June 1999.

[10] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[11] P. A. Ioannou and J. Sun. *Robust Adaptive Control.* Prentice Hall, 1996.

[12] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, second edition, 1995.

[13] I. Kanellakopoulos, P. Kokotovic, and A. S. Morse. Systematic design of adaptive controllers for feedback linearizable systems. *IEEE Trans. on Autom. Control*, 36:1241–1253, 1991.

[14] H. Khalil. *Nonlinear Systems.* Prentice Hall, 2nd edition, 1996.

[15] B. Kim and A. Calise. Nonlinear flight control using neural networks. *Journal of Guidance Control and Dynamics*, 20(1):26–33, 1997.

[16] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design.* Wiley, 1995.

[17] F. L. Lewis, S. Jagannathan, and A. Yesildirek. *Neural Network Control of Robots Manipulators and Nonlinear Systems.* Taylor and Francis, 1999.

[18] D. Mclean. *Automatic Flight Control Systems.* Prentice Hall, 1990.

[19] M. J. Mears and M. M. Polycarpou. Stable local neural control of uncertain systems. *Proceedings of the 38th Conference on Decision and Control*, pages 1691–1696, 1999.

[20] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1):4–27, March 1990.

[21] M. Pachter and P. Chandler. Challenges of autonomous control. *IEEE Contr. Sys. Mag.*, 18(4):92–97, August 1998.

[22] M. Polycarpou and P. Ioannou. Identification and control of nonlinear systems using neural network models: Design and stability analysis. Technical Report 91-09-01, University of Southern California, Dept. Electrical Engineering – Systems, September 1991.

[23] M. M. Polycarpou. Stable adaptive neural control scheme for nonlinear systems. *IEEE Transactions on Automatic Control*, 41(3):447–451, 1996.

[24] M. M. Polycarpou and P. Ioannou. A robust adaptive nonlinear control design. *Automatica*, 32(3):423–427, 1996.

[25] M. M. Polycarpou and M. J. Mears. Stable adaptive tracking of uncertain systems using nonlinearly parameterized on-line approximators. *Inter. Journ. Control*, 70(3):363–384, 1998.

[26] J.-B. Pomet and L. Praly. Adaptive nonlinear regulation: estimation from the lyapunov equation. *IEEE Trans. on Autom. Control*, 37:729–740, 1992.

[27] W. Reigelsperger et al. Application of multivariable control theory to aircraft control laws. Technical Report WL-TR-96-3099, Wright Laboratory, May 1996. Section 10.

[28] G. Rovithakis and M. Christodoulou. Adaptive control of unknown plants using dynamical neural networks. *IEEE Trans. on Systems, Man and Cybernetics*, 24:400–412, March 1994.

[29] N. Sadegh. A perceptron network for functional identification and control of nonlinear systems. *IEEE Trans. on Neural Net.*, 18:982–988, November 1993.

[30] R. M. Sanner and J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863, November 1992.

[31] J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, 1991.

[32] J. Spooner and K. Passino. Stable adaptive control using fuzzy systems and neural networks. *IEEE Transaction on Fuzzy Systems*, 4:339–359, August 1996.

[33] D. Taylor, P. Kokotovic, R. Marino, and I. Kanellakopoulos. Adaptive regulation of nonlinear systems with unmodeled dynamics. *IEEE Trans. on Autom. Control*, 34:405–412, 1989.

[34] D. White and D. Sofge, editors. *Handbook of Intelligent Control.* Van Nostrand Reinhold, 1992.

[35] B. Yao and M. Tomizuka. Adaptive robust control of siso nonlinear systems in a semi-strict feedback form. *Automatica*, 33(5):893–900, 1997.

[36] A. Yesildirek and F. Lewis. Feedback linearization using neural networks. *Automatica*, 31(11):1659–1664, 1995.